

第5章 数据库完整性



沈明玉

合肥工业大学
Hefei University of Technology

计算机与信息学院

第5章 数据库的完整性

主要内容:

- 5.1 实体完整性
- 5.2 参照完整性
- 5.3 用户定义的完整性
- 5.4 完整性约束命名语句
- 5.5 断言
- 5.6 触发器

合肥工业大学
Hefei University of Technology

计算机与信息学院

第5章 数据库的完整性

■ 数据库中数据的完整性:

数据的正确、有效、相容。

■ 数据库的完整性控制:

采取有效方法保证数据库中数据的完整性。

■ DBMS采用的方法:

1. 提供定义完整性约束条件的机制
2. 提供完整性检查的方法
3. 违约处理

合肥工业大学
Hefei University of Technology

计算机与信息学院

第5章 数据库的完整性

5.1 实体完整性

5.1.1 实体完整性定义

CREATE TABLE 中用 PRIMARY KEY 定义:

CREATE TABLE Student

(Sno CHAR(9) PRIMARY KEY, /* 列级定义 */
Sname CHAR(20) NOT NULL,);

CREATE TABLE Student

(Sno CHAR(9),
Sname CHAR(20) NOT NULL,
.....

PRIMARY KEY (Sno)); /* 表级定义 */

5.1 实体完整性

5.1.2 实体完整性检查和违约处理

插入或对主码列进行更新操作时, RDBMS按照实体完整性规则自动进行检查。包括:

1. 检查主码值是否唯一, 如果不唯一则拒绝插入或修改
2. 检查主码的各个属性是否为空, 只要有一个为空就拒绝插入或修改

合肥工业大学
Hefei University of Technology

计算机与信息学院

5.1 实体完整性

全表扫描:

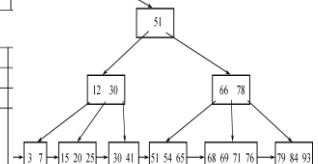
待插入记录

Key1	F21	F31	F41
------	-----	-----	-----

新记录的主码值

索引方法:

Key1	F21	F31
Key2	F22	F32
Key3	F23	F33
...		



合肥工业大学
Hefei University of Technology

计算机与信息学院

第5章 数据库完整性

5.2 参照完整性

5.2.1 参照完整性定义

- 在 CREATE TABLE 中用 FOREIGN KEY 短语定义外码。
- 用 REFERENCES 短语指明外码所参照基本表的主码。

```
CREATE TABLE SC
( Sno CHAR(9) NOT NULL,
  Cno CHAR(4) NOT NULL,
  Grade SMALLINT,
  PRIMARY KEY (Sno, Cno), /*在表级定义实体完整性*/
  FOREIGN KEY (Sno) REFERENCES Student(Sno),
                                /*在表级定义参照完整性*/
  FOREIGN KEY (Cno) REFERENCES Course(Cno)
                                /*在表级定义参照完整性*/
);
```

5.2 参照完整性

5.2.2 参照完整性检查和违约处理

可能破坏参照完整性的情况及违约处理

被参照表（例如Student）	参照表（例如SC）	违约处理
可能破坏参照完整性	插入元组	拒绝
可能破坏参照完整性	修改外码值	拒绝
删除元组	可能破坏参照完整性	拒绝/级连删除/设置为空值
修改主码值	可能破坏参照完整性	拒绝/级连修改/设置为空值

5.2 参照完整性

```
CREATE TABLE SC
( Sno CHAR(9) NOT NULL,
  Cno CHAR(4) NOT NULL,
  Grade SMALLINT,
  PRIMARY KEY (Sno, Cno),
  FOREIGN KEY (Sno) REFERENCES Student(Sno)
    ON DELETE CASCADE /*级联删除SC表中相应的元组*/
    ON UPDATE CASCADE /*级联更新SC表中相应的元组*/
  FOREIGN KEY (Cno) REFERENCES Course(Cno)
    ON DELETE NO ACTION
    /*当删除course 表中的元组造成了与SC表不一致时拒绝删除*/
    ON UPDATE CASCADE
    /*当更新course 表中的cno时, 级联更新SC表中相应的元组*/
); /* 置空值: SET NULL */
```

第5章 数据库的完整性

5.3 用户定义的完整性

CREATE TABLE 时定义: /* 列级、元组级（表级）两种定义方法! */

- 属性值非空（NOT NULL）：通常采用列级定义
- 属性值唯一（UNIQUE）：列级、表级（UNIQUE（属性名））均可
- 检查属性值是否满足一个布尔表达式（CHECK）：两者均可
- 违约处理：违反一条用户定义的规则，均拒绝！

```
CREATE TABLE Student
( Sno CHAR(10) PRIMARY KEY,
  Sname CHAR(8) NOT NULL,
  Ssex CHAR(2) CHECK (Ssex IN ('男', '女')),
  Sage SMALLINT,
  Dno CHAR(2) REFERENCES Dept(Dno),
  UNIQUE (Sname)
);
```

第5章 数据库的完整性

5.4 完整性约束命名子句

CONSTRAINT <完整性约束条件名>

[PRIMARY KEY 短语 | FOREIGN KEY 短语 | CHECK 短语];

```
【例】CREATE TABLE Student (Sno NUMERIC(6)
  CONSTRAINT C1 CHECK (Sno BETWEEN 90000 AND 99999),
  Sname CHAR(20) CONSTRAINT C2 NOT NULL,
  Sage NUMBER(3) CONSTRAINT C3 CHECK (Sage < 30),
  Ssex CHAR(2) CONSTRAINT C4 CHECK (Ssex IN ('男', '女')),
  CONSTRAINT StudentKey PRIMARY KEY(Sno));
```

5.4 完整性约束命名子句

通过命名子句修改完整性定义

【例】修改表Student中的约束条件。

```
ALTER TABLE Student DROP CONSTRAINT C1;
ALTER TABLE Student
  ADD CONSTRAINT C1 CHECK (Sno BETWEEN 90000 AND 99999);

ALTER TABLE Student DROP CONSTRAINT C3;
ALTER TABLE Student
  ADD CONSTRAINT C3 CHECK (Sage < 40);
```

第5章 数据库的完整性

5.5 断言 (Assertion)

■ 定义断言:

```
CREATE ASSERTION <断言名> CHECK <约束规则>
```

- 断言定义了数据库必须满足的约束;
- 断言可以定义涉及多表或聚集操作的更为复杂的完整性约束;
- 复杂的断言会造成很大的处理开销。

```
Create Assertion chk_SC_Num
Check ( 60 >= ( Select count(*) from SC, Course
Where SC.Cno = Course.Cno and Cname = '数据库' ));
```

■ 删除断言:

```
DROP ASSERTION <断言名>;
```

第5章 数据库的完整性

5.6 触发器 (Trigger)

触发器是由一系列SQL语句组成的动作体, 当对数据库做修改(包括插入、删除和更新)时, 它自动被系统执行。

■ 设置触发器机制必须满足的两个条件:

- ① 指明什么条件下触发器被执行, 即触发条件;
- ② 指明触发器执行的动作是什么, 即触发什么。

合肥工业大学 计算机与信息学院

5.6 触发器

■ 触发器的利弊

- ☞ 可以进行更为复杂的检查和操作, 具有更精细和更强大的数据控制能力, 能够保证数据库的一致性;
- ☞ 检测和维护触发器需要很大的开销, 降低了数据库增、删、改的效率!

合肥工业大学 计算机与信息学院

5.6 触发器

◆ 定义触发器

```
CREATE TRIGGER <触发器名>
{ BEFORE | AFTER } <触发事件> ON <表名>
FOR EACH { ROW | STATEMENT }
[ WHEN <触发条件> ]
<触发动作体>;
```

了解: T-SQL与PL/SQL的差异!

合肥工业大学 计算机与信息学院

5.6 触发器

■ 说明:

- 1) 创建者: 表的拥有者
- 2) 触发器名
- 3) 表名: 触发器的目标表
- 4) 触发事件: INSERT、DELETE、UPDATE
- 5) 触发器类型
 - > 行级触发器 (FOR EACH ROW)
 - > 语句级触发器 (FOR EACH STATEMENT)

合肥工业大学 计算机与信息学院

5.6 触发器

[例] 定义一个BEFORE行级触发器, 为教师表Teachers定义完整性规则“讲师的工资不得低于3000元, 如果低于3000元, 自动改为3000元”。

```
CREATE TRIGGER Insert_Or_Update_Sal
BEFORE INSERT OR UPDATE ON Teachers -- 触发事件是插入或更新操作
FOR EACH ROW -- 行级触发器
BEGIN -- 定义触发动作体, 是PL/SQL过程块
IF (:new.Job='讲师') AND (:new.Sal < 3000) THEN
:new.Sal :=3000; -- :old
END IF;
END;
```

合肥工业大学 计算机与信息学院

5.6 触发器

❖ 激活触发器

- ✓ 触发器的执行，是由**触发事件**激活的，并由数据库服务器自动执行。
- ✓ 一个数据表上可能定义了多个触发器。
- ✓ 同一个表上的多个触发器激活时遵循如下的执行顺序：
 - (1) 执行该表上的BEFORE触发器；
 - (2) 激活触发器的SQL语句；
 - (3) 执行该表上的AFTER触发器。

5.6 触发器

❖ 删除触发器

DROP TRIGGER <触发器名> ON <表名>;

- ✓ 触发器必须是一个已经创建的触发器，并且只能由具有相应权限的用户删除。

[例] 删除教师表Teacher上的触发器Insert_Sal。

DROP TRIGGER Insert_Sal ON Teacher;

第5章 数据库完整性

■ 本章思考题：

结合实际系统（如仓库管理系统）理解数据库完整性控制的必要性。

■ 本章作业：

P173 习题2、习题4、习题7